Pokemon Rank Classifier

ECS 171 Machine Learning

Group Leader: Alex D'Souza
Group Members: Catherine Chen, Varun Wadhwa, Shane Kim

Github Repository:
https://github.com/AlexanderDsouza/ECS171GroupProject

Chapter 1

Introduction and background

Pokemon is one of the most popular franchises in the world, with various console games, mobile apps, and a very passionate trading card collecting community. There are over 8 million daily Pokemon Go players, over 480 million lifetime console game sales, and over 3.7 billion pokemon cards sold in the 2020-2021 fiscal year. The biggest pokemon tournament is the Pokemon World Championship with over 110,000 USD in prize money, so every advantage a player can get is extremely valuable. A big advantage that players can get is being able to categorize pokemon with regards to their strength level. There are a couple of reasons this is advantageous:

1. Accurately measuring the overall strength of a team

Tournaments ban certain pokemon that are considered too "strong" so a player's team will be made up of pokemon from various strength levels. Therefore, to know the overall strength of a player's roster, knowing the individual strength of each pokemon is crucial. Being able to accurately measure the overall strength of a team also allows a player to know how their roster stacks up against an opposing player's roster. This allows players to strategically choose pokemon to fill their team as well as determine movesets

2. Efficient Training

A Pokemon's base stats can be improved via a concept known as Effort Values by battling other pokemon. Therefore knowing the strength level of an individual pokemon can allow a player to determine which pokemon need to be trained more.

3. Predictions

Knowing the objective strength level of a pokemon, allows a player to predict what moves their opponent will make against theirs. If a pokemon is “strong” , moves that lower its attack or defense stats may be used. If a pokemon is weak, one shot moves may be preferred.

The current issue around categorizing pokemon is that there is no single statistic that determines if a pokemon is strong. For example, one pokemon may have strong attack statistics and one pokemon may have strong defense statistics so in order to determine which pokemon is stronger, various statistics must be considered. Even if an individual is able to manually categorize all 1021 pokemon using their own methodology, new pokemon are frequently released, so manually evaluating each pokemon and comparing them to create strength tiers is not feasible. Machine learning poses an elegant solution to this issue, as being able to produce strength level predictions for new pokemon based on previous generation data without manually having to evaluate and categorize new pokemon saves valuable time.

Chapter 2

Literature Review

1. Previous Attempts:

- **Breddan** attempted to predict or categorize attributes of Pokemon, investigating relationships between different statistics such as type, HP, attack, defense, special attack, special defense, speed, and total strength.
- Explored whether distinct groups of Pokemon could be formed based on these statistics to aid team diversification.
- Attempted to predict one statistic of a Pokemon based on others to anticipate new Pokemon's stats using educated guesses.

2. Methodology Used:

- Employed k-means classification and Random Forest Regression techniques.

3. Findings:

- Successfully created 5 distinct groups of Pokemon using k-means classification with speed and HP data.
- Predicted total strength using Random Forest Regression with a maximum tree depth of 11, achieving an accuracy of 95

Chapter 3

Dataset Description

The dataset used in this project is a CSV file obtained from Kaggle, generated by scraping pokeapi.co through Python. It comprises 1017 rows and 18 columns, featuring the following attributes:

- **Columns:**
  - ID of the Pokemon
  - Name
  - Rank (strength level)
  - Generation
  - Evolution chain
  - Primary type
  - Secondary type
  - HP
  - Attack
  - Defense
  - Special attack
  - Special defense
  - Speed
  - Total stats
  - Height
  - Weight
  - Abilities

For our model, we decided to consider only numerical data as input features, excluding categorical features such as primary type, secondary type, abilities, evolution information, and others. We made this choice because including these categorical features via one-hot encoding would significantly increase the model's complexity and training time.

Our target variable, Rank, serves as a measure of strength and encompasses four categories: legendary, mythical, baby, and ordinary. We identified 'baby' and 'ordinary', as well as 'mythical' and 'legendary', as redundant labels concerning strength levels. Consequently, we categorized all 'baby' Pokemon as 'ordinary', and 'mythical' as 'legendary'.

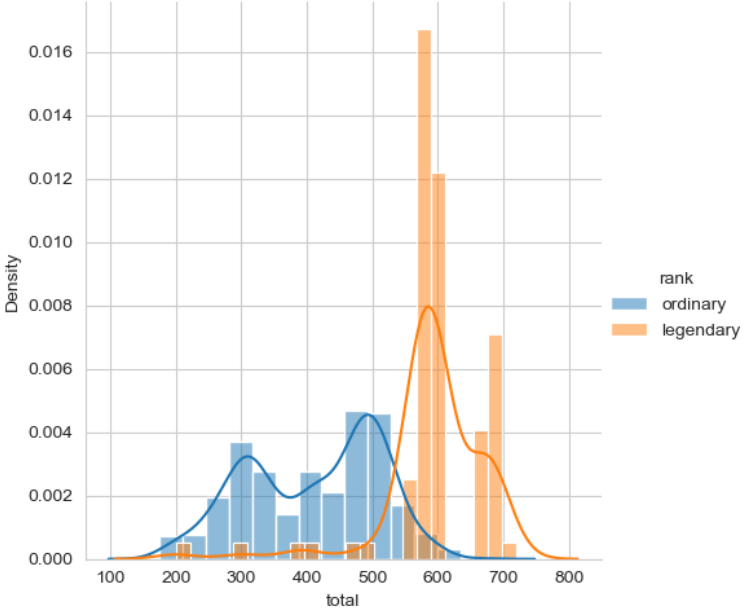

Figure 3.1: Histogram of the Dataset

This section provides a description of the dataset used in our study. To illustrate the characteristics of the dataset, we include a histogram plot shown in Figure 3.1. As you can see at around 550 in total stats, legendary Pokemon and ordinary Pokemon make a distinct split. When we saw this, we decided that a linear SVM model would be perfect for our dataset as you can clearly draw a line right in between that mark and it could split the classes pretty accurately.

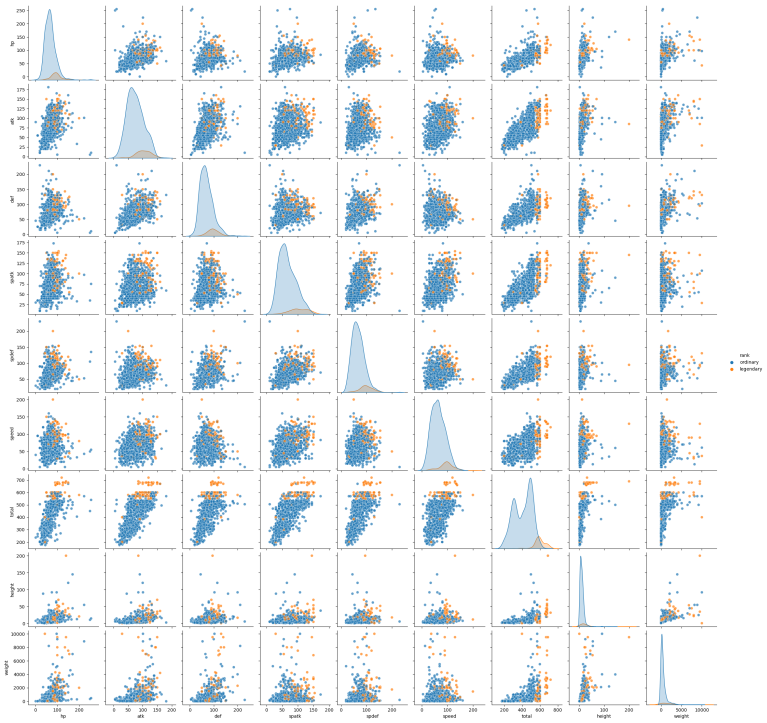

Figure 3.2: Pair Plot of the Dataset

In the pairplot we can also see that in the total pairplots, we can easily cut the classes and group them with a line. This also led us to believe SVM would be extremely good.

Chapter 4

Proposed Methodology

4.1 Model selection

For our Pokemon rank classifier, we opted to utilize Logistic Regression and Support Vector Machine (SVM) models due to their distinct advantages and capabilities in handling this classification task.

Logistic Regression

Pros:

- Well-suited for binary classification problems like ours.
- Offers probabilistic interpretation of results.
- Generally less prone to overfitting.

Cons:

- Assumes linear relationship between features and the log-odds of the outcome.
- May not perform optimally with non-linear data distributions.

Support Vector Machine (SVM)

Pros:

- Effective in high-dimensional spaces.
- Versatile due to different kernel functions.

- Robust against overfitting in high-dimensional spaces.

Cons:

- Computationally intensive with larger datasets.
- Choice of kernel and parameters crucial for performance.

4.1.1 User Prediction Interface

Our classifier allows users to predict whether a chosen Pokemon is legendary or not. Initially, we considered enabling users to specify the Effort Values (EVs) of the Pokemon. However, for simplicity and consistency with our test data, users can select a Pokemon, and our system will randomly distribute their EVs within allowable constraints.

4.1.2 Data Augmentation

To enhance our dataset and balance class representation, we augmented the dataset by generating additional data points. We increased the representation of 'legendary' Pokemon to ensure robustness in the training phase.

4.1.3 Model Training and Evaluation

Data Preprocessing: We preprocessed the data, removing unnecessary features and ensuring numerical consistency.

Model Training: We trained both Logistic Regression and SVM models on the augmented dataset.

Model Evaluation: Using train-test splitting, we evaluated the models' performance and obtained classification reports for each model.

4.1.4 Model Persistence

Upon successful training and evaluation, we persisted the SVM model as 'Pokemon_Predictor.pkl', enabling the deployment of the trained model for future predictions.

Chapter 5

Experimental Results

Chapter 6

Conclusion and Discussion

Chapter 7

References